Software contracts or: how I learned to stop worrying and love releasing

Seb Rose
Agenda

Test doubles
Software contracts
Contract testing
Consumer driven contracts
Pact & Pact Broker
Q & A
Agenda

- Test doubles
- Software contracts
- Contract testing
- Consumer driven contracts
- Pact & Pact Broker
- Q & A
Test automation pyramid

How much of the application the test exercises

Number of tests

http://claysnow.co.uk/architectural-alignment-and-test-induced-design-damage-fallacy/

@sebrose

http://smartbear.com
Dependencies
Dependencies replaced

A

Test double

Test double

Test double
What could possibly go wrong?
Agenda

Test doubles

Software contracts

Contract testing

Consumer driven contracts

Pact & Pact Broker

Q & A

@sebrose

http://smartbear.com
Design by contract

Contract
• an agreement between client and supplier

Characteristics
• expect some benefits
• incur some obligations
Explicit contract
Implicit contract
Implicit contract
interface ICalculate
{
    public int int Add(int a, int b);
}

Agenda

Test doubles
Software contracts
Contract testing
Consumer driven contracts
Pact & Pact Broker
Q & A
Collaboration tests -> Production

Collaboration

Test double

Common approach
Success!

Collaboration Test

Production

Test double
Fail

Collaboration Test

Production

Test double
Contract test

Contract Test

Test
double
Systematic contract testing

• Collaboration tests make assumptions about the contract

• Contract tests try to justify those assumptions

JB Rainsberger, via GOOS mailing list, “Unit-test mock/stub assumptions rots”
15 March 2012
Contract test through an interface

Contract Test

Test double
“Don’t mock what you don’t own”

Joe Walnes
From first principles ...

Contract Test

Interface

Test double
Our explicit contract

```csharp
public interface IRevenueProvider
{
    decimal GetRevenue(int customerId);
}
```
What’s our implicit contract?

- Pass in a valid customer ID, return a decimal value
- Can we say anything about the magnitude of the returned value?
- Pass in an invalid customer ID…. then what?
Documenting the contract

[TestFixture]
public abstract class RevenueProviderContract
{
    private IRevenueProvider revenueProvider;

    protected abstract IRevenueProvider GetRevenueProvider();
}
private IRevenueProvider revenueProvider;

protected abstract
       IRevenueProvider GetRevenueProvider();

[SetUp]
public void setup()
{
    revenueProvider = GetRevenueProvider();
}

[Test]
public void valid_customer_id_id()
public class ProductionContract : RevenueProviderContract
{
    public ProductionContract() {}

    protected override IRevenueProvider GetRevenueProvider()
    {
        return new ProductionRevenueProvider();
    }
}

http://smartbear.com
public class TestDoubleContract : RevenueProviderContract
{
    public TestDoubleContract() {}

    protected override IReRevenueProvider GetRevenueProvider()
    {
        return new TestRevenueProvider();
    }
}
Agenda

Test doubles
Software contracts
Contract testing
Consumer driven contracts
Pact & Pact Broker
Q & A
Provider contracts

▪ Closed and complete  Provider contracts express a service's business function capabilities in terms of the complete set of exportable elements available to consumers, and as such are closed and complete with respect to the functionality available to the system.

▪ Singular and authoritative  Provider contracts are singular and authoritative in their expression of the business functionality available to the system.

▪ Bounded stability and immutability  A provider contract is stable and immutable for a bounded period and/or locale. Provider contracts typically use some form of versioning to differentiate differently bounded instances of the contract.

https://martinfowler.com/articles/consumerDrivenContracts.html

@sebrose
Consumer driven contracts (CDC)

▪ **Open and incomplete** Consumer contracts are open and incomplete with respect to the business functionality available to the system. They express a subset of the system's business function capabilities in terms of the consumer's expectations of the provider contract.

▪ **Multiple and non-authoritative** Consumer contracts are multiple in proportion to the number of consumers of a service, and each is non-authoritative with regard to the total set of contractual obligations placed on the provider. Consumers may evolve at different rates.

▪ **Bounded stability and immutability** Like provider contracts, consumer contracts are valid for a particular period of time and/or location.
How do you agree a contract?
How do you agree a contract?

Provider team

Consumer team

@sebrose

http://smartbear.com
How do you agree a contract?
How do you agree a contract?
How do you agree a contract?
Agenda

Test doubles
Software contracts
Contract testing
Consumer driven contracts
Pact & Pact Broker
Q & A
Continuous Integration (CI)

CI  Staging  Prod

A3 ———— A2 ———— A1
CI - consumer & provider

CI

Staging

Prod

C_3

C_2

C_1

P_3

P_2

P_1
Simplifying micro-service testing

Pact provides a mechanism for creating a contract between a service consumer and a service provider, and then providing the tools to validate that the consumer and provider adhere to the contact independently of each other.
Demo

https://github.com/tdshipley/pact-workshop-dotnet-core-v1
Pact - key points

• Consumer creates contracts using Pact DSL
• Pact creates mock HTTP server
• Running the tests creates a Pact file
• Provider uses Pact file to verify compatibility
• Provider may offer “backdoor” interface
Microservices

Cl

V_{32}
W_{3}
X_{9}
Y_{15}
Z_{84}

Staging

V_{29}
W_{2}
X_{8}
Y_{14}
Z_{73}

Prod

V_{27}
X_{7}
Y_{9}
Z_{72}

@sebrose
http://smartbear.com
https://www.youtube.com/watch?v=79GKBYSqMlo
If you can’t deploy services independently, you don’t have microservices.

You have a distributed monolith

Beth Skurrie
<table>
<thead>
<tr>
<th>Consumer version</th>
<th>Provider version</th>
<th>Verification result</th>
</tr>
</thead>
<tbody>
<tr>
<td>11</td>
<td>31</td>
<td>success</td>
</tr>
<tr>
<td>12</td>
<td>31</td>
<td>failure</td>
</tr>
<tr>
<td>12</td>
<td>32</td>
<td>success</td>
</tr>
<tr>
<td>13</td>
<td>32</td>
<td>success</td>
</tr>
</tbody>
</table>

https://www.youtube.com/watch?v=79GKBYSqMlo
can-i-deploy
Pact broker - key points

• Pacts are published by Consumer
• Pacts are fetched by Provider
• Results are stored in the “Matrix”
• “Matrix” supports independent deployment
Take aways

• Increases confidence
• Reduces need for integration tests
• Speeds up development

• No substitute for communication
• Does not replace all other testing
Seb Rose

Twitter: @sebrose
Blog: http://cucumber.io/blog
E-mail: seb@smartbear.com

http://bddbooks.com